Formal Verification of Finger Print ATM Transaction through Real Time Constraint Notation (RTCN)

Vivek K. Singh[1], Tripathi S. P.[2], R. P. Agarwal[3] and Singh J. B.[4]

[1] School of Computer Engineering & Information Technology, Shobhit University, Research Scholar Meerut, U.P. 250110, India

[2] Department of Computer Science & Engineering, Gautam Buddh Technical University, IET Lucknow, U.P. 226021, India

[3] School of Computer Engineering & Information Technology, Shobhit University, Professor Meerut, U.P. 250110, India

[4] School of Computer Engineering & Information Technology, Shobhit University, Professor Meerut, U.P. 250110, India

Abstract

In this paper we propose the Formal Verification of existing models like in banking sector ie ATM Transaction through biometric (Finger Print) with the help of Real Time Constraint Notation. Finger print recognition is most popular and commonest method of using the biometrics. In the finger print technology, the uniqueness of epidermis of fingers is utilized for identification of user. The user has to keep its finger on a sensory pad, which reads the ridges of epidermis of finger and try to match it with available data of the finger with the bank.

Sequence Diagrams (SDs), Finite State Machine (FSM) have proven useful for describing transaction-oriented systems, and can form a basis for creating state charts. However, Finger Print ATM system require special support for branching, state information, and composing SDs.

Keywords: *SD_Sequence Diagram, FPI_Finger Print Impression, DB_Data Base,OCL_Object Constraint Language, FSM_Finite State Machine*

1. Introduction

Traditionally, access to secure areas or sensitive information has been controlled by possession of a particular artifact (such as a card or key) and/or knowledge of a specific piece of information such as a Personal Identification Number (PIN) or a password. Today, many people have PINs and passwords for a multitude of devices, from the car radio and mobile phone, to the computer, web-based services and their bank information. Herein lies a major difficulty involving the trade-off between usability, memorability and security[17]. Methods for increasing security, such as regularly changing PINs and passwords, increasing their length, ensuring they do not form words and ensuring all are different, makes them more difficult to remember and, therefore, error-prone. Alternatives to the traditional Personal Identification Number (PIN) have also been investigated for instance using pictures (finger print) instead of numbers [18]. Of course, traditional methods rely upon the assumption that the artifact (such as key or card) will be in the possession of the rightful owner and that the information to activate it will be kept secret. Unfortunately, neither of these assumptions can be wholly relied upon. If people are permitted to choose their own passwords they tend to select ones which are easily guessed. People tend to choose ones that are related to their everyday life [17]. They choose passwords which are easy to remember, and, typically, easily predicted, or they

change all PINs to be the same. Also, people are often lax about the security of this information and may deliberately share the information, say with a spouse or family member, or write the PIN down and even keep it with the card itself. Biometric techniques [19] may ease many of these problems: they can confirm that a person is actually present (rather than their token or passwords) without requiring the user to remember anything. In this paper, we explore how to use UML sequence diagrams to support the needs of finger print ATM verification system. First, we review methods for composing sequence diagrams that support flexible finger print ATM modeling. Then, we show how determining required information content can be represented as finite state machine to guarantee correct, cohesive diagrams. A generic approach is described; with supporting finger print ATM verification system incorporating data, state, and timing information. Finally, the more commonly discussed transaction processing model is revisited to illustrate system differences.

2. Biometric Approach – ATM transaction through Finger Print recognition

A *biometric system* is essentially a pattern recognition system that recognizes a person by determining the authenticity of a specific physiological and / or behavioral characteristic possessed by that person. An important issue in designing a practical biometric system is to determine how an individual is recognized. Depending on the application context, a biometric system may be called either a *verification* system or an *identification* system [16]:

1. A verification system authenticates a person's identity by comparing the captured biometric characteristic with her own biometric template(s) pre-stored in the system. It conducts one-to-one comparison to determine whether the identity claimed by the individual is true. A verification system either rejects or accepts the submitted claim of identity.
2. An identification system recognizes an individual by searching the entire template database for a match. It conducts one-to-many comparisons to establish the identity of the individual. In an identification system, the system establishes a subject's identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity.

The term *authentication* is also frequently used in the biometric field, sometimes as a synonym for verification; actually, in the information technology language, authenticating a user means to let the system know the user identity regardless of the mode (verification or identification).

The banking and financial sector has adopted this system wholeheartedly because of its robustness and the advantages it provides in cutting costs and making processes more streamlined. The technology started out as a novelty however due exigencies in the banking sector characterized by decreasing profits it became a necessity. The use of Biometric ATM's based on finger print recognition technology has gone a long way in improving customer service by providing a safe and paperless banking environment.

Identification of right user by the use of face recognition technology is the latest form of biometric ATMs. Identification based on the different walk style while entering in ATMs is used in gait based ATMs.

Benefits of biometric technology: Since biometric technology can be used in the place of PIN codes in ATMs, its benefits mostly accrues to rural and illiterate masses who find it difficult to use the keypad of ATMs. Such people can easily put their thumbs on the pad available at ATMs machines and proceed for their transactions.

Biometric technology provides strong authentication, as it uses the unique features of body parts. This helps reduce the chances of occurring frauds in ATM usage.

Though use of biometric technology has its high cost implications to banks, several other costs of conventional ATMs like re-issuance of password, helpdesk etc will be reduced, which will be a positive factor for banks to go for biometric ATMs.

3. Terminology Used

3.1 Scenario, Sequence Diagrams, State Chart & Message Sequence Charts:

A scenario is a sequence of events that occurs during one particular execution of a system. A scenario describes a way to use a system to accomplish some function [5]. Scenarios can be expressed in many forms, textual and graphical, informal and formal. Sequence diagrams emphasize temporal ordering of events, whereas collaboration diagrams focus on the structure of interactions between objects. Each may be readily translated into the other. State chart diagrams represent the behavior of entities capable of dynamic behavior by specifying its response to the receipt of event in stances.

Typically, it is used for describing the behavior of classes, but state charts may also describe the behavior of other model entities such as us e-cases, actors, subsystems, operations, or methods. Message sequence charts constitute an attractive visual formalism that is widely used to capture system requirements during the early design stages in domains such as ATM transaction via fingerprint recognition [15].

3.2 Composition of Scenarios:

A crucial challenge in describing formal verification of fingerprint ATM recognition is the composition of scenarios. In order to be adequately expressive, sequence diagrams must reflect the structures of the programs they represent. In this paper, we survey approaches to modeling execution structures and transfer of control, and select a method that lends itself to Fingerprint Verification System.

Our objective is to refine a model that utilizes sequential, conditional, iterative, and concurrent execution. As many ideas exist, our task is to determine which are appropriate for Fingerprint Verification System. Hsia et al. [5] discusses a process for scenario analysis that includes conditional branching. Glinz [2] includes iteration as well. Koskimies et al.[8] and Systa [13] present a tool that handles "algorithmic scenario diagrams" - sequence diagrams with sequential, iterative, conditional and concurrent behavior. We use elements of each, for a combined model that allows sequential, conditional, iterative, and concurrent behavior.

Another objective is to model transfer of control through sequence diagram composition. The main decision to make is where to annotate control information. One approach is to include composition information in individual diagrams.

3.3 Finite State Machines (FSM):

*By Dr. Matt Stallmann & Suzanne Balik: "*A *finite-state machine (FSM)* is an abstract model of a system (physical, biological, mechanical, electronic, or software)".

A *finite state machine* (FSM) is a mathematical model of a system that attempts to reduce the model complexity by making simplifying assumptions. Specifically, it assumes

1. The system being modeled can assume only a finite number of conditions, called *states.*
2. The system behavior within a given state is essentially identical.
3. The system resides in states for significant periods of time.
4. The system may change these conditions only in a finite number of well defined ways, called *transitions*.
5. Transitions are the response of the system to *events*.
6. Transitions take (approximately) zero time.

3.4 Object Constraint Language (OCL):

The Object Constraint Language (OCL) is an expression language that enables one to describe constraints on object – oriented models and other object modeling artifacts.

A constraint is a restriction on one or more values of (part of) of an object oriented model or system. OCL is the part of the Unified Modeling Language (UML), the OMG (Object Management Group, a consortium with a membership of more than 700 companies. The organization's goal is to provide a common framework for developing applications using object-oriented programming techniques) standard for object – oriented analysis and design.

OCL has been used in a wide variety of domains, and this has led to the identification of some under – specified areas in the relationship between OCL and UML.

OCL can be used for a number of different purposes:

1. to specify invariants on classes and types in the class model
2. to specify type invariants for Stereotypes
3. to describe pre- and post- conditions on Operations and Methods
4. to describe guard
5. as a navigation language
6. to specify constraint on operations

In OCL, UML operation semantics can be expressed using pre and post condition constraints – The pre condition says what must be true for the operation to meaningfully execute – The post condition expresses what is guaranteed to be true after execution completes

1. About the return value
2. About any state changes (e.g. instance variables)

4. Proposed Formal Verification of Finger Print ATM Transaction through Real Time Constraint Notation (RTCN) : -

Now we are going to demonstrate the formal verification of ATM transaction through Fingerprint Verification Model with the help of Sequence Diagram their corresponding Finite State Machine and their corresponding Real Time Constraint Notation with the help of Object Constraint Language (OCL). There are four objects exchanging messages: the user, the ATM, the consortium, and the bank. In this example, State charts are generated for the ATM object only. The scenarios share the same initial condition.

1. Through Sequence Diagrams (SD's):

Case 1: Transaction Fail due to mismatch Finger Print Impression (FPI) at server site database:

In this case, ATM transaction fails due to mismatch of finger print through finger print database (DB) file server site:

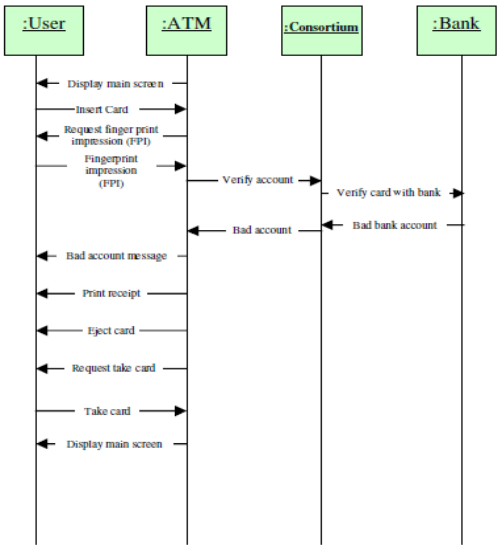

Figure 1 Sequence Diagram for Fingerprint Verification ATM, Case1: Mismatch Fingerprint(FPI)

Case 2: Transaction success due to match of Finger Print Impression (FPI) at server database:

In this case, ATM transaction is successfully processed due to accurate match of finger print through finger print database (DB) file server site:

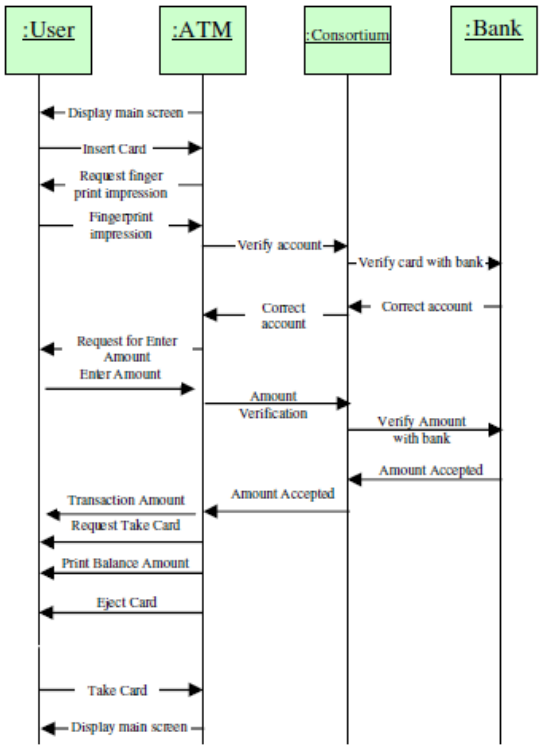

Figure 2 Sequence Diagram for Fingerprint Verification ATM, Case2: Correct FPI, Successful Transaction

2. Finite State Machine corresponding to Sequence Diagrams (SD's):

The above two cases can be represented with help of their corresponding Finite State machine as :

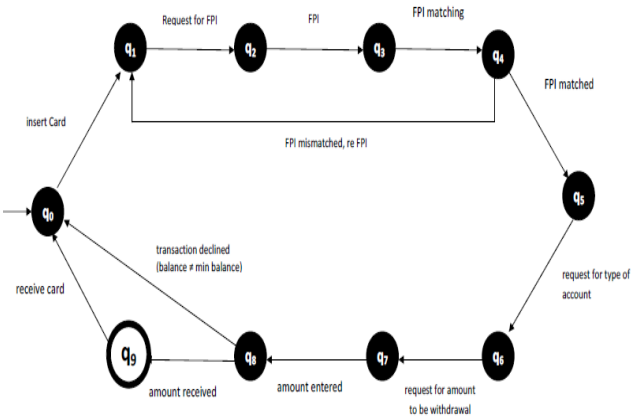

Figure 3 : Finite State Machine for ATM Transaction by FPI

Fig 3 representing the Finite State Machine (FSM) for the transaction through ATM after the verification of Finger Print Impression(FPI). FSM has ten states represented as q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, where q_0 is an initial state and q_9 is the final state of the FSM. We can define the transition function (δ) for the proper working FSM as shown in Fig 3 as follows:

1. $\delta(q_0,$ insert card$) = q_1$
2. $\delta(q_1,$ request for FPI$) = q_2$
3. $\delta(q_2,$ FPI$) = q_3$
4. $\delta(q_3,$ FPI matching$) = q_4$
5. 5.1 $\delta(q_4,$ FPI matched$) = q_5$
   5.2 $\delta(q_4,$ FPI mismatched, re FPI$) = q_1$
6. $\delta(q_5,$ request for type of account$) = q_6$
7. $\delta(q_6,$ request for amount to be withdrawal$) = q_7$
8. $\delta(q_7,$ amount entered$) = q_8$
9. 9.1 $\delta(q_8,$ transaction declined$) = q_0$
   9.2 $\delta(q_8,$ amount received$) = q_9$ - Final State
10. $\delta(q_9,$ receive card$) = q_0$ - Initial State

3. Real Time Constraint Notation corresponding to Sequence Diagrams (SD's) (Case 1 & 2) with the help of Object Constraint Language(OCL):

We are proposing the Object Constraint Language (OCL) notation for user's transaction through ATM after finger print verification as :

User :: Finger Print Impression(FPI)

pre : Finger Print Identification

post :
Request Transaction.Saving Account ->reject (not (FPI = User))
or
Request Transaction.Saving Account ->reject((FPI = User))
and
(Request Transaction and Saving Account @ pre + Account Balance > = Minimum Balance)
and
(Request Transaction and Saving Account @ post + Account Balance > = Account Balance – Withdrawal Amount)

5. Conclusion:

We presented a methodology that guarantees sufficient sequence diagram information to generate correct Statecharts. We converted sequence diagrams to the respective Finite State Machine (FSM) and also give the Object Constraint Language (OCL), pre / post conditions for transaction process through ATM. When state, message preconditions, and timing information are included in the FSM & OCL notations seems to be sufficient to guarantee determinism for the Fingerprint ATM Verification we discussed. We have also examined diagram composition and information content to assess adequacy for Fingerprint ATM Verification.

References

[1] Douglass, B. Doing Hard Time. Addison-Wesley, 99.

[2] Glinz, M. An Integrated Formal Model of Scenarios Based on Statecharts. In *Proceedings of the 5th European Software Engineering Conference (ESEC 95)*, Sitges, Spain, 1995, pp.254-271.

[3] Harel, D. Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, vol.8, no.3, 1987, pp. 231-274.

[4] Hitz, M., and G. Kappel. Developing with UML - Some Pitfalls and Workarounds. *UML '98 - The Unified Modeling Language*, Lecture Notes in Computer Science 1618, Springer-Verlag, 1999, pp. 9-20.

[5] Hsia, P. et al. Formal Approach to Scenario Analysis. *IEEE Software*, vol.11, no.2, 1994, pp.33-41.

[6]ITU-T. Recommendation Z.120. ITU Telecommunication Standardization Sector, Geneva, Switzerland, May 1996.

[7] Khriss, I., M. Elkoutbi, and R. Keller. Automating the Synthesis of UML StateChart Diagrams from Multiple Collaboration Diagrams. *UML '98 - The Unified Modeling Language,* Lecture Notes in Computer Science 1618, Springer-Verlag, 1999, pp. 132-147.

[8] Koskimies, K., T. Systa, J. Tuomi, and T. Mannisto. Automated Support for Modeling Software. *IEEE Software*, vol.15, no.1, 1998, pp. 87-94.

[9] Leue, S., L. Mehrmann, and M. Rezai. Synthesizing Software Architecture Descriptions from Message Sequence Chart Specifications. In *Proceedings of the 13th IEEE International Conference on Automated Software Engineering*, Honolulu, Hawaii, 1998, pp. 192-195.

[10] Li, X. and J. Lilius. Checking Compositions of UML Sequence Diagrams for Timing Inconsistency. In *Proceedings of the Seventh Asia-Pacific Software Engineering Conference (APSEC 2000)*, Singapore, 2000, pp. 154-161.

[11] Louden, K. Compiler Construction : Principles and Practice. PWS Publishing Company, 1997.

[12] Some, S., R. Dssouli, and J. Vaucher. From Scenarios to Timed Automata: Building Specifications from User Requirements. In *Proceedings of the 1995 Asia-Pacific Software Engineering Conference*, Australia, 1995, pp. 48-57.

[13] Systa, T. Incremental Construction of Dynamic Models for Object-Oriented Software Systems. *Journal of Object-Oriented Programming*, vol.13, no.5, 2000, pp. 18-27.

[14] Unified Modeling Language Specification, Version 1.3, 1999. Available from the Object Management Group. http://www.omg.com.

[15] Elizabeth Latronico and Philip Koopman: Representing Embedded System Sequence Diagrams As A Formal Language

[16] D. Maltoni, D. Maio, Handbook of Fingerprint Recognition, Springer, 2003

[17] Adams, A. and Chang, S.Y. An investigation of keypad interface security. *Information & Management,* 24, 53-59, 1993.

[18] De Angeli, A., Coutts, M. Coventry, L., Johnson, G.I, Cameron D., and Fischer M. VIP: a visual approach to user authentication. *Proceedings of the Working Conference on Advanced Visual Interfaces AVI 2002*, ACM Press, pp. 316-323, 2002

[19] Ashbourn, J. *Biometrics. Advanced Identity Verification*. Springer Verlag, London, 2000.

Acknowledgments

Singh, Vivek thanks Darbari, Manuj without his essential guidance this research paper would not have been possible .

Singh, Vivek is currently working as Assistant Professor in the Department of I. T. at BBDNITM, Lucknow. He has over 10 years of teaching & experience. Having done his B.Tech in Computer Science & Engineering from Purvanchal University in 2001, M.Tech from U.P.Technical University, Lucknow in 2006, he is pursuing his Ph.D. from Shobhit University, Meerut.

Tripathi, S.P. (Dr.) is currently working as Associate Professor in Department of Computer Science & Engineering at I.E.T. Lucknow. He has over 30 years of experience. He has published numbers of papers in referred Journals.

Agarwal, R. P. (Dr.) is currently working as Professor & Ditrector in School of CE&IT at Shobhit University, Meerut. He has 40 years of teaching experience and has published number of papers in referred Journals.

Singh, J.B. (Dr.) is currently working as Dean Students Welfare at Shobhit University, Meerut. He has 38 years of teaching experience and has published number of papers in referred Journals.